

**This Page Is Inserted by IFW Operations  
and is not a part of the Official Record**

## **BEST AVAILABLE IMAGES**

**Defective images within this document are accurate representations of the original documents submitted by the applicant.**

**Defects in the images may include (but are not limited to):**

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

MENU

SEARCH

INDEX

1/1



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: 08087411

(43) Date of publication of application: 02.04.1996

(51) Int. Cl.

G06F 9/38  
G06F 9/38

(21) Application number: 06223677

(71) Applicant:

(22) Date of filing: 19.09.1994

(72) Inventor:

FUJITSU LTD

MORIWAKI SHINICHI

YANAGIDA MASAHIRO

FUJIOKA SHUNTARO

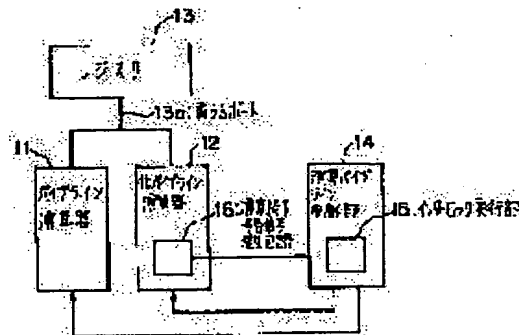
OTA HIDENOBU

## (54) METHOD AND DEVICE FOR PIPELINE OPERATION

## (57) Abstract:

**PURPOSE:** To efficiently write the operation results of respective computing elements in a register by the addition of a simple circuit and to improve an arithmetic processing performance of a method and device which perform arithmetic processing by plural computing elements including a pipeline computing element and a non-pipeline computing element while one write port is shared.

**CONSTITUTION:** This device is equipped with the pipeline computing element 11 which can perform pipeline operation, the non-pipeline computing element 12 which can not perform the pipeline operation, and a register 13 whose write port 13a are shared by those computing elements 11 and 12; when operations are carried out while the operation results of those computing elements 11 and 12 are written in the register 13, the execution of a stage where instructions to the pipeline computing element 11 are decoded is interlocked when the write stage of the operation result of the pipeline computing element 11 is overlapped with the write stage of the operation result of the non-pipeline computing element 12.



LEGAL STATUS

【発行国】日本国特許庁(JP)  
 【公報種別】公開特許公報  
 【公開番号】特開平8-87411 平成8年(1996)4月2日

【発明の名称】パイプライン演算方法およびパイプライン演算装置

【国際特許分類第6版】

(IPC,識別,分冊,庁内No,技術箇所)

G06F 9/38,310,J,,

G06F 9/38,370,C,,

【審査請求】未請求

【FI】

(FI,識別,分冊,庁内No,技術箇所)

G06F 9/38,310,J,,

G06F 9/38,370,C,,

【請求項の数】6

【出願形態】OL

【全頁数】19

【出願番号】特願平6-223677 平成6年(1994)9月19日

【出願人】富士通株式会社(000005223)

神奈川県川崎市中原区上小田中1015番地

【発明者】森脇 真一

神奈川県川崎市中原区上小田中1015番地 富士通株式会社内

・柳田 昌宏

神奈川県川崎市中原区上小田中1015番地 富士通株式会社内

・藤岡 俊太郎

神奈川県川崎市中原区上小田中1015番地 富士通株式会社内

・大田 秀信

神奈川県川崎市中原区上小田中1015番地 富士通株式会社内

【代理人】[弁理士]真田 有

【要約】

【目的】

本発明は、パイプライン演算器と非パイプライン演算器とを含む複数の演算器により1つの書込ポートを共有し演算処理を実行する方法および装置に関し、簡素な回路の追加により、各演算器の演算結果を効率よくレジスタに書き込めるようにして、演算処理性能の向上をはかることを目的とする。

【構成】

パイプライン動作可能なパイプライン演算器11と、パイプライン動作不可能な非パイプライン演算器12と、これらの演算器11, 12により書込ポート13aを共有されるレジスタ13とをそなえ、各演算器11, 12による演算結果をレジスタ13に書き込みながら演算を実行する際に、パイプライン演算器11による演算結果の書込ステージと非パイプライン演算器12による演算結果の書込ステージとが重なる場合、パイプライン演算器11に対する命令を解読するステージの実行をインターロックするように構成する。

【特許請求の範囲】

【請求項1】

パイプライン動作可能なパイプライン演算器と、パイプライン動作不可能な非パイプライン演算器と、これらのパイプライン演算器および非パイプライン演算器により書込ポートを共有されるレジスタとをそなえ、前記の各演算器による演算結果を該レジスタに書き込みながら演算を実行するパイプライン演算方法であって、

該パイプライン演算器による演算結果の書込ステージと該非パイプライン演算器による演算結果の書込ステージとが重なる場合、該パイプライン演算器に対する命令を解読するステージの実行をインターロックすることを特徴とする、パイプライン演算方法。

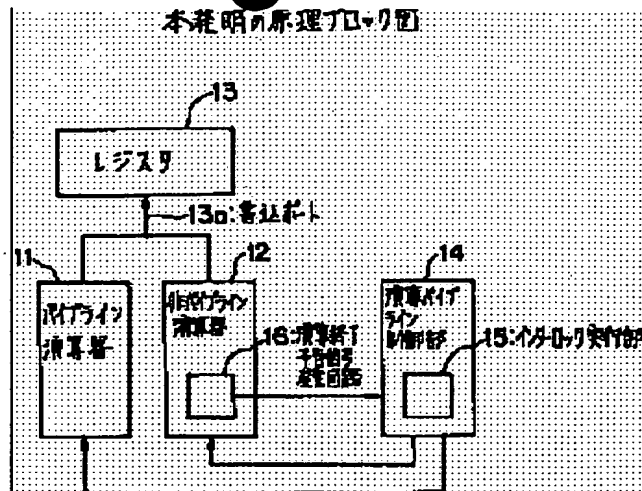
【請求項2】

該非パイプライン演算器が演算を終了する旨を予告・通知する演算終了予告信号を生成し、該演算終了予告信号に応じて、該パイプライン演算器に対する命令を解読するステージの実行を1ステージ分だけインターロックすることを特徴とする、請求項1記載のパイプライン演算方法。

【請求項3】

前記演算終了予告信号が、該非パイプライン演算器のn-m番目[mは該パイプライン演算器の演算実行ステージ数、n(>m)は該非パイプライン演算器の演算実行ステージ数]の演算実行ステージの実行時に生成されることを特徴とする、請求項2記載のパイプライン演算方法。

【請求項4】



パイプライン動作可能なパイプライン演算器と、  
パイプライン動作不可能な非パイプライン演算器と、  
これらのパイプライン演算器および非パイプライン演算器により書込ポートを共有されるレジスタと、  
これらのパイプライン演算器および非パイプライン演算器における演算動作を制御する演算パイプライン制御部とをそなえ、  
該演算パイプライン制御部により前記の各演算器における演算動作を制御し前記の各演算器による演算結果を該レジスタに書き込みながら演算を実行するパイプライン演算装置であって、  
該演算パイプライン制御部に、  
該パイプライン演算器による演算結果の書込ステージと該非パイプライン演算器による演算結果の書込ステージとが重なる場合、該パイプライン演算器に対する命令を解読するステージの実行をインターロックするインターロック実行部がそなえられていることを特徴とする、パイプライン演算装置。

**【請求項5】**

該非パイプライン演算器に、該非パイプライン演算器が演算を終了する旨を予告・通知する演算終了予告信号を生成して該演算パイプライン制御部へ出力する演算終了予告信号発生回路がそなえられ、  
該インターロック実行部が、該演算終了予告信号発生回路からの演算終了予告信号に応じて、該パイプライン演算器に対する命令を解読するステージの実行を1ステージ分だけインターロックすることを特徴とする、請求項4記載のパイプライン演算装置。

**【請求項6】**

該演算終了予告信号発生回路が、該非パイプライン演算器の $n-m$ 番目[ $m$ は該パイプライン演算器の演算実行ステージ数、 $n(>m)$ は該非パイプライン演算器の演算実行ステージ数]の演算実行ステージの実行時に、前記演算終了予告信号を生成することを特徴とする、請求項5記載のパイプライン演算装置。

**【発明の詳細な説明】****【0001】**

(目次)

産業上の利用分野

従来の技術(図6～図11)

発明が解決しようとする課題

課題を解決するための手段(図1)

作用(図1)

実施例(図2～図5)

発明の効果

**【0002】****【産業上の利用分野】**

本発明は、パイプライン演算器(例えば乗算器等)と非パイプライン演算器(例えば除算器等)とを含む複数の演算器により1つの書込ポートを共有して、演算処理を実行するパイプライン演算方法およびパイプライン演算装置に関する。

**【0003】****【従来の技術】**

一般に、複数の演算器からの演算結果を浮動小数点レジスタ[以下、FR(Floating point Register)という場合もある]に書き込む手法としては、(1)FR側に演算器の数だけの書込ポートを設け書込タイミングの制御を行なうことなく各演算器の演算結果をいつでもFRに書き込む手法と、(2)1つの書込ポートを複数の演算器で共有し書込タイミングの制御を行ないながら各演算器の演算結果をFRに書き込む手法とがある。

**【0004】**

ところで、近年の計算機では、命令を実行するために、例えば図6に示すようなパイプライン動作を行なっている。

なお、図6において、2は命令キュー、3は命令レジスタ、4はデコーダ(DEC)、5はパイプライン制御回路、6は浮動小数点レジスタ(FR)、9はパイプライン演算器で、これらの各構成要素については、図8により詳細に説明する。

**【0005】**

パイプライン動作とは、前の命令の実行完了を待たずに引き続く命令の実行を開始するような動作で、複数の段階に分解可能な処理が、各段階に対応した複数の処理機構へと配分されて処理され、処理されていく一つのデータが同一処理装置を複数回経路するような動作をしない。

そのパイプライン動作は、図6に示すように、命令選択ステージSf、レジスタ読出ステージFf、演算実行ステージE1f～Emf、書込ステージWfから構成されている。

ここで、命令選択ステージSfは命令を選択するステージ、レジスタ読出ステージFfは命令を解読しレジスタからデータを読み出すステージ、演算実行ステージE1f～Emfは演算を実行するステージ

〔図6にはパイプライン演算器9の演算サイクル数(演算実行ステージ数)がmの場合、つまりmステージに亘って演算を実行する場合が示されている〕、書込ステージWfは演算結果をレジスタに書き込むステージである。

【0006】

そして、パイプライン動作では、まず、最初の命令選択ステージSfで、実行すべき命令を、パイプライン制御回路5により命令キュー2から選択して命令レジスタ3に書き込み、次のレジスタ読出ステージFfで、命令選択ステージSfで選択された命令をデコーダ4により解読し、その解読結果に基づいてFR6からデータを読み出してくる。

【0007】

この後、演算器9における演算実行ステージE1f~Emfで、読み出されたデータを用いて実際に演算を実行し、演算実行ステージEmfで得られた最終的な演算結果を、書込ステージWfでFR6に書き込む。

以上のような動作を、1サイクル(1ステージ)ずつずらした状態で、複数ステージを並列的に実行(オーバーラップ実行)することにより、毎サイクル、データを入力し、毎サイクル、演算結果を出力する。

このような動作が可能な演算器がパイプライン演算器9である。

【0008】

一方、演算器としては、上述のようなパイプライン動作可能なパイプライン演算器9(例えば乗算器等)のほか、例えば除算器等のように、複数ステージの並列的な実行(オーバーラップ実行)ができない、つまり、パイプライン動作が不可能な非パイプライン演算器もある。

このような非パイプライン演算器では、図7に示すように、演算が開始されると、演算が終了するまでの $(n+1) \cdot \tau$ の間、レジスタ読出ステージFfを実行できず次のデータを入力することができない。

なお、nは非パイプライン演算器の演算サイクル数(演算実行ステージ数)、 $\tau$ は1サイクル(1ステージ)に要する時間(制御周期)であり、図7において、7は非パイプライン演算器である。

【0009】

従って、パイプライン演算器9のようなパイプライン動作を行なうことは不可能である。

一般的に、除算器は非パイプライン演算器で、演算時間が非常に長いので、命令として出現頻度が少ない。

そこで、一般に、図8や図10に示すように、除算器等の非パイプライン演算器7とパイプライン動作が可能な演算器9とについては、FR6の書込ポート6aを共有することが行なわれている。

【0010】

しかし、書込ポート6aを共有した場合、2つの演算器7、9は同時にFR6へ書込を行なうことができないため、従来、以下の2つの手法が用いられている。

(a) 非パイプライン演算器7が動作している間、パイプライン演算器9を動作させない。

【0011】

(b) 非パイプライン演算器7の書込ステージWfとパイプライン演算器9の書込ステージWfとが重なった時だけ、パイプライン演算器9の書込ステージWfを1 $\tau$ だけ遅らせる。

前者(a)の手法を適用したパイプライン演算装置の構成を図8に示す。

この図8において、1は演算パイプライン制御部で、この演算パイプライン制御部1は、図示しない制御部から演算命令を受け取り、演算器7、9およびFR6を含む演算ユニット全体のパイプラインの制御を行なうもので、命令キュー2、命令レジスタ3、デコーダ(DEC)4およびパイプライン制御回路5を有して構成されている。

【0012】

ここで、命令キュー2は、図示しない制御部から送られてくる演算命令を保持するものであり、命令レジスタ3は、パイプライン制御回路5により選択されて命令キュー2から読み出された演算命令を一時的に格納するものである。

また、デコーダ4は、命令レジスタ3に格納された演算命令を解読し、その解読結果をFR6や各演算器7、9へレジスタ番号、命令信号、演算開始信号として送るものである。

【0013】

さらに、パイプライン制御回路5は、命令キュー2に対して命令選択信号を発行するとともに、非パイプライン演算器7が動作している間、デコーダ4に対してレジスタ読出ステージFfのインターロック信号を発行し、演算ユニットのパイプラインを制御するものである。

そして、FR6は、演算パイプライン制御部1で指定されたアドレスのデータを読み出し格納するとともに、各演算器7、9による演算結果を格納するためのものである。

【0014】

また、非パイプライン演算器7は、除算器等で、演算パイプライン制御部1(デコーダ4)から演算開始信号を受け取ると演算を開始するものであり、パイプライン演算器9は、乗算器等で、前述した通りのパイプライン動作可能なもので、演算パイプライン制御部1(デコーダ4)から演算開始信号を受け取ると演算を開始するものである。

【0015】

上述の構成により、図9に示すように、非パイプライン演算器7が演算を開始した後に、書込ポート6aを共有するもう一方のパイプライン演算器9でも演算を開始しようとした場合、命令選択ステージSfの終了後、パイプライン制御回路5からデコーダ4へインターロック信号が出力され、レジスタ読出ステージFfでインターロックがかかり、パイプライン演算器9は演算を開始しない。

#### 【0016】

そして、非パイプライン演算器7による演算が終了し、その演算結果を書込ステージWfでFR6に書き込むと、インターロック信号の出力が停止されてレジスタ読出ステージFfのインターロック状態が解除され、パイプライン演算器9による演算が開始される。

従って、パイプライン演算器9は、非パイプライン演算器7の演算サイクル数(演算実行ステージ数)を $n$ とすると、 $(n+1) \cdot \tau$ の間、待機することになる。

#### 【0017】

次に、後者(b)の手法を適用したパイプライン演算装置の構成を図10に示す。

図10に示すように、このパイプライン演算装置も図8に示したものとほぼ同様に構成されているが、図10に示すパイプライン演算装置のパイプライン制御回路5に、インターロック信号発生回路5Aがそなえられている。

このインターロック信号発生回路5Aは、非パイプライン演算器7の書込ステージWfとパイプライン演算器9の書込ステージWfとが重なった時に、パイプライン演算器9に対して書込ステージWfのインターロック信号を発行するものである。

#### 【0018】

上述の構成により、図11に示すように、非パイプライン演算器7が演算を行なっている際に、書込ポート6aを共有するもう一方のパイプライン演算器9も演算を実行し、パイプライン演算器9の書込ステージWfが、非パイプライン演算器7の書込ステージWfのタイミングと重なった時、パイプライン演算器9の書込ステージWfで、インターロック信号発生回路5Aからパイプライン演算器9に対してインターロック信号が発行される。

#### 【0019】

パイプライン演算器9は、インターロック信号を受け取ると、インターロック状態になり、書込ステージWfの実行を停止し、非パイプライン演算器7がFR6に演算結果を書き込むまでの $1\tau$ の間だけ待機してから、インターロック状態を解除し書込ステージWfを実行し、パイプライン演算器9による演算結果をFR6に書き込む。

#### 【0020】

##### 【発明が解決しようとする課題】

しかしながら、前者(a)の手法を適用した従来のパイプライン演算装置では、非パイプライン演算器7が演算を行ない書込ステージWfでFR6への書込を終えるまでの最大 $(n+1) \cdot \tau$ の間、パイプライン演算器9を使用することができなくなるため、演算処理性能が大幅に低下するという課題があった。

#### 【0021】

これに対して、後者(b)の手法を適用した従来のパイプライン演算装置では、前者の装置についての課題は解消されるが、書込ステージWfでパイプライン演算器9を停止させるためには、パイプライン演算器9における各パイプラインを同時に停止させる機構が必要になる。

このような停止機構を追加すると、回路が複雑になり物量が増加するなどの課題があった。

#### 【0022】

本発明は、このような課題に鑑み創案されたもので、簡素な回路を追加するだけで、書込ポートを共有する複数の演算器による演算結果を効率よくレジスタに書き込めるようにして、演算処理性能の向上をはかったパイプライン演算方法およびパイプライン演算装置を提供することを目的とする。

#### 【0023】

##### 【課題を解決するための手段】

図1は本発明の原理ブロック図で、この図1において、11はパイプライン動作可能なパイプライン演算器、12はパイプライン動作不可能な非パイプライン演算器、13はパイプライン演算器11および非パイプライン演算器12により書込ポート13aを共有され各演算器11、12による演算結果を書き込まれるレジスタ、14はパイプライン演算器11および非パイプライン演算器12における演算動作を制御する演算パイプライン制御部である。

#### 【0024】

そして、演算パイプライン制御部14には、インターロック実行部15がそなえられている。

このインターロック実行部15は、パイプライン演算器11による演算結果の書込ステージと非パイプライン演算器12による演算結果の書込ステージとが重なる場合、パイプライン演算器11に対する命令を解釈するステージの実行をインターロックするものである(請求項1、4)。

#### 【0025】

また、非パイプライン演算器12に、この非パイプライン演算器12が演算を終了する旨を予告・通知する演算終了予告信号を生成して演算パイプライン制御部14へ出力する演算終了予告信号発生回路16をそなえ、インターロック実行部15が、演算終了予告信号発生回路16からの演算終了予

告信号に応じて、パイプライン演算器11に対する命令を解読するステージの実行を1ステージ分だけインターロックするように構成する(請求項2, 5)。

**【0026】**

このとき、演算終了予告信号発生回路16は、非パイプライン演算器12の $n-m$ 番目( $m$ はパイプライン演算器11の演算実行ステージ数、 $n(>m)$ は非パイプライン演算器12の演算実行ステージ数)の演算実行ステージの実行時に、演算終了予告信号を生成する(請求項3, 6)。

**【0027】****【作用】**

図1により上述した本発明のパイプライン演算方法およびパイプライン演算装置では、演算パイプライン制御部14により各演算器11, 12における演算動作を制御し、各演算器11, 12による演算結果をレジスタ13に書き込みながら演算を実行する。

**【0028】**

そして、パイプライン演算器11による演算結果の書込ステージと非パイプライン演算器12による演算結果の書込ステージとが重なる場合には、インターロック実行部15により、パイプライン演算器11に対する命令を解読するステージの実行がインターロックされる。

これにより、パイプライン演算器11に対する命令が解読されず、パイプライン演算器11による演算処理の実行が一時的に延期される。

従って、従来のごとく書込ステージでパイプライン演算器11の動作を停止させることなく、非パイプライン演算器12による演算結果の書込ステージと非パイプライン演算器12による演算結果の書込ステージとの実行タイミングをずらして、各演算結果をレジスタ13へ書き込むことができる(請求項1, 4)。

**【0029】**

また、非パイプライン演算器12が演算を終了する前には、その旨を予告する演算終了予告信号が、演算終了予告信号発生回路16により生成され、演算パイプライン制御部14に通知される。そして、演算終了予告信号に応じ、演算パイプライン制御部14のインターロック実行部15により、パイプライン演算器11に対する命令を解読するステージの実行が1ステージ分だけインターロックされる。

**【0030】**

これにより、命令解読後のパイプライン演算器11による演算処理の実行が1ステージ分だけ遅延され、結果として、非パイプライン演算器12による演算結果の書込ステージと非パイプライン演算器12による演算結果の書込ステージとの実行タイミングが1ステージ分だけずれることになり、各演算結果をレジスタ13へ順次書き込むことができる(請求項2, 5)。

**【0031】**

このとき、演算終了予告信号発生回路16により、演算終了予告信号が、非パイプライン演算器12の $n-m$ 番目の演算実行ステージの実行時、即ち、非パイプライン演算器12が演算を終了する( $m+1$ )ステージ前に生成される。

つまり、この演算実行ステージの実行時にパイプライン演算器11に対する命令を解読するステージを実行した場合、パイプライン演算器11による演算結果の書込ステージと非パイプライン演算器12による演算結果の書込ステージとが重なることになる。

**【0032】**

従って、上述のタイミングで生成された演算終了予告信号に応じて、パイプライン演算器11に対する命令を解読するステージの実行を1ステージ分だけ確実にインターロックすることができ、前述の通り、命令解読後のパイプライン演算器11による演算処理の実行が1ステージ分だけ遅延され、非パイプライン演算器12による演算結果および非パイプライン演算器12による演算結果をレジスタ13へ順次書き込むことができる(請求項3, 6)。

**【0033】****【実施例】**

以下、図面を参照して本発明の実施例を説明する。

図2は本発明の一実施例としてのパイプライン演算装置を示すブロック図であり、この図2において、21は演算パイプライン制御部で、この演算パイプライン制御部21は、図示しない制御部から演算命令を受け取り、後述する除算器27、乗算器29および浮動小数点レジスタ(FR)26を含む演算ユニット全体のパイプラインの制御を行なうもので、命令キュー22、命令レジスタ23、デコーダ(DEC)24およびパイプライン制御回路25を有して構成されている。

**【0034】**

ここで、命令キュー22は、図示しない制御部から送られてくる演算命令を保持するものであり、命令レジスタ23は、パイプライン制御回路25により選択されて命令キュー22から読み出された演算命令を一時的に格納するものである。

また、デコーダ24は、命令レジスタ23に格納された演算命令を解読し、その解読結果をFR26や除算器27、乗算器29へレジスタ番号、命令信号、演算開始信号として送るものである。

**【0035】**

さらに、パイプライン制御回路25は、命令キュー22に対して命令選択信号を発行するほか、後述

するインターロック信号発生回路30を有しており、演算ユニットのパイプラインを制御するものである。

そして、FR26は、演算パイプライン制御部21で指定されたアドレスのデータを読み出し格納するとともに、除算器27、乗算器29により書込ポート26aを共有され、これらの除算器27、乗算器29による演算結果を格納するものである。

【0036】

また、除算器27は、パイプライン動作不可能な非パイプライン演算器で、演算パイプライン制御部21(デコーダ24)から演算開始信号を受け取ると演算を開始するものであり、乗算器29は、パイプライン動作可能なパイプライン演算器で、やはり演算パイプライン制御部21(デコーダ24)から演算開始信号を受け取ると演算を開始するものである。

【0037】

そして、本実施例の除算器27には、演算終了予告信号発生回路28がそなえられている。除算器27の演算実行ステージ数(演算サイクル数)を $n$ 、乗算器29の演算実行ステージ数(演算サイクル数)を $m$ ( $<n$ )、1サイクル(1ステージ)に要する時間(制御周期)を $\tau$ とした場合、演算終了予告信号発生回路28は、除算器27の $n-m$ 番目の演算実行ステージの実行時、即ち、除算器27が演算を終了する $(m+1) \cdot \tau$ 前に、除算器27が演算を終了する旨を予告・通知する演算終了予告信号を生成して演算パイプライン制御部21へ出力するものである。

【0038】

一方、演算パイプライン制御部21のパイプライン制御回路25には、上述した演算終了予告信号発生回路28からの演算終了予告信号を受けて動作するインターロック信号発生回路(インターロック実行部)30がそなえられている。

このインターロック信号発生回路30は、除算器27による演算結果の書込ステージWfと乗算器29による演算結果の書込ステージWfとが重なる場合、乗算器29に対する命令を解読するステージFfの実行をインターロックするためのものである。

【0039】

より詳細に説明すると、インターロック信号発生回路30は、除算器27の演算終了予告信号発生回路28から演算終了予告信号を受けた場合、その時点でレジスタ読出ステージ(乗算器29に対する命令の解読処理を含むステージ)Ffを実行すると、除算器27による演算結果の書込ステージWfと乗算器29による演算結果の書込ステージWfとが重なるものと判断し、デコーダ24に対してレジスタ読出ステージFfのインターロック信号を発行することにより、レジスタ読出ステージFfの実行を1ステージ( $1\tau$ )分だけインターロックするようになっている。

【0040】

上述のごとく構成された本実施例のパイプライン演算装置におけるパイプライン制御回路25および演算終了予告信号発生回路28の動作を、それぞれ図3(a)、(b)を参照しながら説明する。まず、図3(a)に示すフローチャート(ステップS1~S4)により本実施例のパイプライン制御回路25の基本的な動作を説明すると、このパイプライン制御回路25は、所定サイクル $1\tau$ 毎に、命令キュー22に対してその命令キュー22内に格納されている命令を選択して命令レジスタ23へ出力するための命令選択信号を出力する(ステップS1、命令選択ステージSf)。

【0041】

そして、ステップS1による命令選択動作の次サイクルで、インターロック発生回路30により除算器27から演算終了予告信号を受けたか否かを判定し(ステップS2)、受けていない場合には、ステップS1に戻り、前述と同様にして命令選択動作を行なう。

これに対して、ステップS2で除算器27から演算終了予告信号を受けたと判定した場合、インターロック発生回路30からデコーダ24に対して、レジスタ読出ステージFfをインターロックするためのFfインターロック信号を $1\tau$ 間だけ出力する(ステップS3)。

これにより、デコーダ24はパイプライン演算器29に対する命令の解読動作を停止し、パイプライン演算器29に対してコマンドバリッド(命令信号、演算開始信号)を発行しなくなり、パイプライン演算器29はレジスタ読出ステージFfで $1\tau$ 間だけインターロックされる。

【0042】

レジスタ読出ステージFfのインターロックを行なったステージの次のステージでは、インターロック発生回路30からデコーダ24へのFfインターロック信号の出力を停止して、レジスタ読出ステージFfのインターロックを解除し、デコーダ24による命令解読動作を許可する(ステップS4)。

この後、パイプライン制御回路25による処理はステップS1へ移行する。

また、レジスタ読出ステージFfを1ステージだけインターロックしたパイプライン演算器29のパイプラインは、それ以降、通常通りの動作(演算実行ステージE1f~Emfおよび書込ステージWf)を実行する。

【0043】

また、図3(b)に示すフローチャート(ステップS11、S12)により本実施例の演算終了予告信号発生回路28の基本的な動作を説明すると、この演算終了予告信号発生回路28は、所定サイクル $1\tau$ 毎に、除算器27の動作状態を監視し、除算器27の動作ステージが、 $n-m$ 番目の演算実行ステージE( $n-m$ )fになったか否かを判定する(ステップS11)。



**[0044]**

除算器27の演算実行ステージE(n-m)fの実行時に、乗算器29のレジスタ読出ステージFfが同時に実行された場合、除算器27による演算結果の書込ステージWfと乗算器29による演算結果の書込ステージWfとが重なってしまい、FR26の書込ポート26aを共有している状態では、いずれの書込処理も行えなくなってしまう。

**[0045]**

そこで、本実施例では、前述したように、演算終了予告信号発生回路28により、除算器27の動作ステージがn-m番目の演算実行ステージE(n-m)fになったか否か、つまり、除算器27の動作ステージが演算を終了する(m+1)ステージ前になったか否かを判定し、該当ステージとなった場合(ステップS11でYES判定の場合)、(m+1)ステージ後には除算器27の演算が終了する旨を予告・通知する演算終了予告信号を、演算終了予告信号発生回路28から演算パイプライン制御部21(パイプライン制御回路25)に対して発行している(ステップS12)。

**[0046]**

そして、この演算終了予告信号を受けたパイプライン制御回路25では、図3(a)にて前述した通りステップS2~S4の処理が実行され、乗算器29に対するレジスタ読出ステージFfを1τ間だけインターロックすることにより、レジスタ読出ステージFf後の乗算器29による演算処理の実行が1ステージ分だけ遅延される。

その結果、図4、図5に示すように、除算器27による演算結果の書込ステージWfと乗算器29による演算結果の書込ステージWfとの実行タイミングが1ステージ分だけずれることになり、各演算結果をFR26へ順次書き込むことができる。

**[0047]**

次に、上述のように動作するパイプライン制御回路25および演算終了予告信号発生回路28をもつ本実施例のパイプライン演算装置全体の動作を、図4を参照しながら説明する。

なお、図4は、除算器27の演算実行ステージ数をn、乗算器29の演算実行ステージ数をm(<n)とした場合の動作を示すタイムチャートである。

**[0048]**

まず、最初の命令選択ステージSfで、命令キュー22内に格納されている複数の演算命令の中から、パイプライン制御回路25から送られてくる命令選択信号によって次に実行すべき命令を選択し、命令レジスタ23に格納する。

次のレジスタ読出ステージFfで、命令レジスタ23内に格納されている命令を演算パイプライン制御部21内のデコーダ24によって解釈し、その解釈結果に基づき、レジスタ信号をFR26へ発行して演算に必要なアドレスデータをFR26から除算器27もしくは乗算器29に対して読み出す。

また、これと同時に命令信号および演算開始信号を除算器27もしくは乗算器29に対して発行することにより、演算(演算実行ステージ)が開始される。

**[0049]**

命令キュー22から選択された命令が除算命令の場合には、レジスタ読出ステージFfで、デコーダ24により除算命令が解釈され、そのデコード結果に基づき、演算に必要なアドレスデータをFR26から除算器27に対して読み出し、同時に除算器27に対して命令信号および演算開始信号が発行され、除算演算が開始される。

これ以降、パイプライン演算回路25により命令キュー22から乗算命令が選択されると、乗算器29による演算処理が、除算器27による演算処理と並行して、パイプライン動作として実行される。

**[0050]**

そして、図4に示すように、除算器27のn-(m+1)番目の演算実行ステージE[n-(m+1)]fで、パイプライン制御回路25から発行される命令選択信号によって、命令キュー22内に格納されている命令の中から、書込ポート26aを共有する乗算器29に対する演算命令が選択され命令レジスタ23に格納された場合、その乗算器29に対する演算命令をそのまま実行すると、除算器27の書込ステージWfと乗算器29の書込ステージWfとが重なってしまう。

**[0051]**

そこで、本実施例では、図3(b)にて前述した通り、除算器27が演算を終了する(m+1)ステージ前に、つまり、除算器27のn-m番目の演算実行ステージE(n-m)fで、除算器27の演算終了予告信号発生回路28から演算終了予告信号を、演算パイプライン制御部21(パイプライン制御回路25)に対して発行している。

**[0052]**

この演算終了予告信号を受けたパイプライン制御回路25では、図3(a)にて前述した通り、インターロック信号発生回路30によりデコーダ24に対してFfインターロック信号が1τ(1ステージ)間だけ発行され、選択された命令の解釈を実行しない。

その結果、乗算器29に対して演算信号および演算開始信号が発行されなくなり、乗算器29には、図4に示すように、レジスタ読出ステージFfで、1τ(1ステージ)の間、インターロックがかかり、レジスタ読出ステージFf以降の演算実行ステージE1f~Emfの実行がすべて1τずつ遅延される。

**[0053]**

従って、乗算器29による演算結果の書込ステージWfが、除算器27による演算結果の書込ステー

ジWfよりも1ステージ分だけ遅れて実行されることになり、これらの書込ステージWfが重ならず、各演算結果をFR26へ順次書き込むことができる。

なお、図5は、上述のように動作するパイプライン制御回路25および演算終了予告信号発生回路28をもつ本実施例のパイプライン演算装置全体の動作を、演算実行ステージ数としてより具体的な数値を設定して示すもので、基本的な動作は図4により説明したものと全く同様である。この図5では、除算器27の演算実行ステージ数 $n$ を38、乗算器29の演算実行ステージ数 $m$ を5とした場合の動作が示されている。

#### 【0054】

この図5に示す具体例では、除算器27の32[ $=n-(m+1)$ ]番目の演算実行ステージE32fで、命令キュー22内に格納されている命令の中から乗算器29に対する演算命令が選択され命令レジスタ23に格納された場合、その乗算器29に対する演算命令をそのまま実行すると、除算器27の書込ステージWfと乗算器29の書込ステージWfとが重なってしまう。

#### 【0055】

そこで、前述した通り、除算器27が演算を終了する6( $=m+1$ )ステージ前に、つまり、除算器27の33( $=n-m$ )番目の演算実行ステージE33fで、除算器27の演算終了予告信号発生回路28から演算終了予告信号をパイプライン制御回路25に対して発行し、インターロック信号発生回路30によりデコーダ24に対してFfインターロック信号が1 $\tau$ (1ステージ)間だけ発行され、選択された命令の解釈を実行しない。

#### 【0056】

その結果、乗算器29に対して演算信号および演算開始信号が発行されなくなり、乗算器29には、図5に示すように、レジスタ読出ステージFfで、1 $\tau$ (1ステージ)の間、インターロックがかかり、レジスタ読出ステージFf以降の演算実行ステージE1f~E5fの実行がすべて1 $\tau$ ずつ遅延されて、乗算器29による演算結果の書込ステージWfが、除算器27による演算結果の書込ステージWfよりも1ステージ分だけ遅れて実行されることになり、これらの書込ステージWfが重ならず、各演算結果をFR26へ順次書き込むことができる。

#### 【0057】

このように、本発明の一実施例によれば、乗算器29による演算結果の書込ステージWfと除算器27による演算結果の書込ステージWfとが重なる場合、乗算器29のレジスタ読出ステージ(命令解読ステージ)Ffの実行が1 $\tau$ 間だけインターロックされる。

従って、従来のごとく書込ステージWfでパイプライン演算器である乗算器29の動作を停止させることなく、除算器27による演算結果の書込ステージWfと乗算器29による演算結果の書込ステージWfとの実行タイミングがずれ、各演算結果をFR26へ書き込めるので、極めて簡素な回路(演算終了完了信号発生回路28)を除算器27に追加するだけで、書込ポート26aを共有する複数の演算器27、29による演算結果を効率よくFR26に書き込むことができ、演算処理性能を大幅に向上することができるのである。

#### 【0058】

なお、上述した実施例では、パイプライン演算器が乗算器で非パイプライン演算器が除算器である場合について説明したが、本発明は、これに限定されるものでなく、パイプライン演算器は加算器、減算器等であってもよいし、非パイプライン演算器はパイプライン動作不可能に構成されたものであればどのような演算器であってもよい。

#### 【0059】

##### 【発明の効果】

以上詳述したように、本発明のパイプライン演算方法およびパイプライン演算装置によれば、パイプライン演算器による演算結果の書込ステージと非パイプライン演算器による演算結果の書込ステージとが重なる場合には、パイプライン演算器に対する命令を解読するステージの実行をインターロックすることにより、パイプライン演算器による演算処理の実行が一時的に延期される。

#### 【0060】

従って、非パイプライン演算器による演算結果の書込ステージと非パイプライン演算器による演算結果の書込ステージとの実行タイミングをずらして、各演算結果をレジスタへ書き込めるので、簡素な回路を追加するだけで、書込ポートを共有する複数の演算器による演算結果を効率よくレジスタに書き込むことができ、演算処理性能を大幅に向上できる効果がある(請求項1、4)。

#### 【0061】

また、非パイプライン演算器が演算を終了する前にその旨を予告する演算終了予告信号を生成し、その演算終了予告信号に応じて、パイプライン演算器に対する命令を解読するステージの実行を1ステージ分だけインターロックすることにより、命令解読後のパイプライン演算器による演算処理の実行が1ステージ分だけ遅延されて、非パイプライン演算器による演算結果の書込ステージと非パイプライン演算器による演算結果の書込ステージとの実行タイミングが1ステージ分だけずれ、各演算結果を効率よくレジスタに書き込むことができ、演算処理性能の向上に寄与することになる(請求項2、5)。

#### 【0062】

このとき、非パイプライン演算器の $n-m$ 番目[mはパイプライン演算器の演算実行ステージ数、 $n$

は非パイプライン演算器の演算実行ステージ数]の演算実行ステージの実行時に、演算終了予告信号を生成することにより、その演算終了予告信号に応じて、パイプライン演算器に対する命令解読ステージの実行を1ステージ分だけ確実にインターロックすることができ、前述と同様の効果を得ることができる(請求項3, 6)。

【図面の簡単な説明】

【図1】

本発明の原理ブロック図である。

【図2】

本発明の一実施例としてのパイプライン演算装置を示すブロック図である。

【図3】

(a)は本実施例のパイプライン制御回路の動作を説明するためのフローチャート、(b)は本実施例の演算終了予告信号発生回路の動作を説明するためのフローチャートである。

【図4】

本実施例の動作を説明するためのタイムチャートである。

【図5】

本実施例の動作を説明するためのタイムチャートである。

【図6】

一般的なパイプライン動作を説明するためのタイムチャートである。

【図7】

一般的な非パイプライン動作を説明するためのタイムチャートである。

【図8】

パイプライン演算器と非パイプライン演算器とにより1つの書込ポートを共有するパイプライン演算装置の一例を示すブロック図である。

【図9】

図8に示すパイプライン演算装置の動作を説明するためのタイムチャートである。

【図10】

パイプライン演算器と非パイプライン演算器とにより1つの書込ポートを共有するパイプライン演算装置の他例を示すブロック図である。

【図11】

図10に示すパイプライン演算装置の動作を説明するためのタイムチャートである。

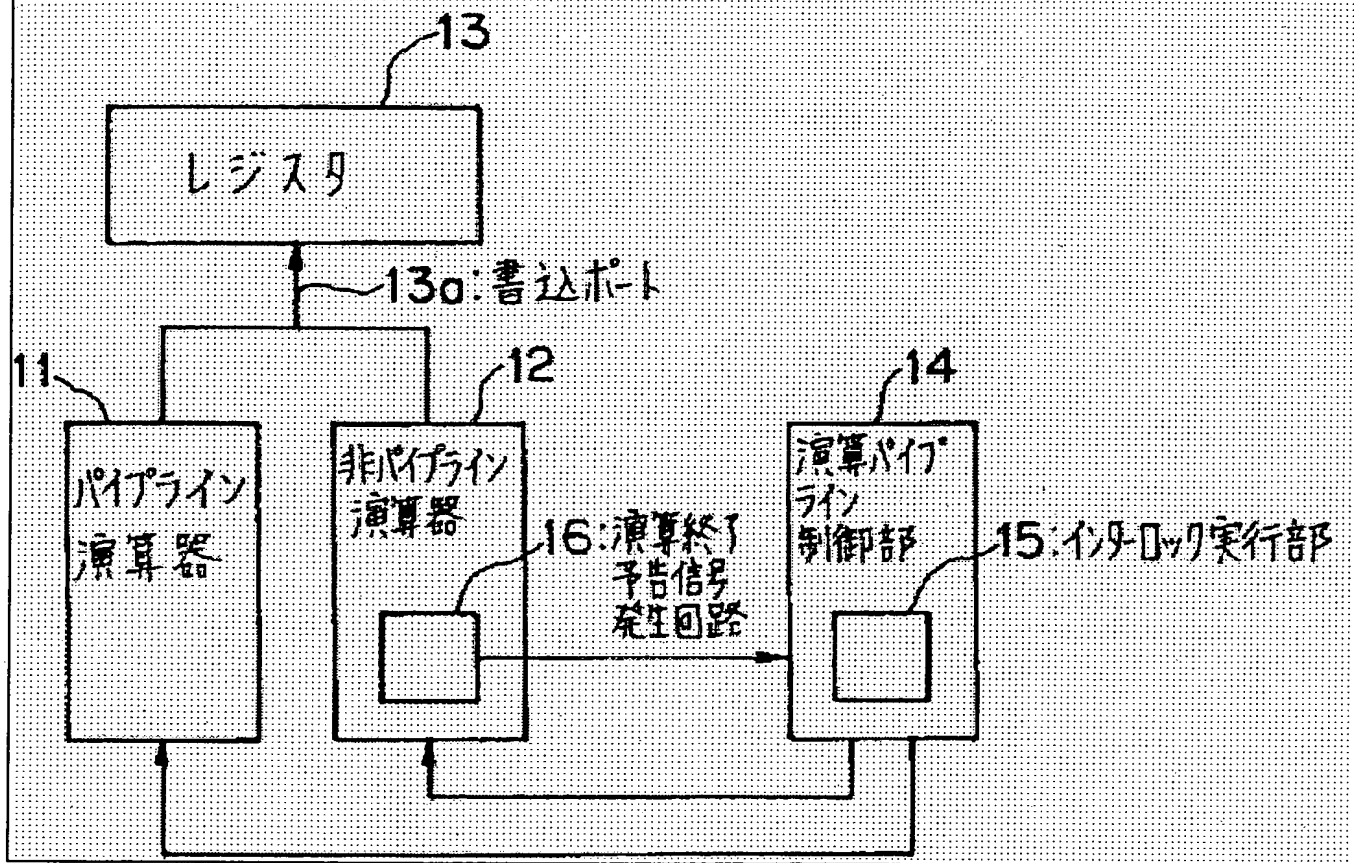
【符号の説明】

- 11 パイプライン演算器
- 12 非パイプライン演算器
- 13 レジスタ
- 13a 書込ポート
- 14 演算パイプライン制御部
- 15 インターロック実行部
- 16 演算終了予告信号発生回路
- 21 演算パイプライン制御部
- 22 命令キュー
- 23 命令レジスタ
- 24 デコーダ(DEC)
- 25 パイプライン制御回路
- 26 浮動小数点レジスタ(FR)
- 26a 書込ポート
- 27 除算器(非パイプライン演算器)
- 28 演算終了予告信号発生回路
- 29 乗算器(パイプライン演算器)
- 30 インターロック信号発生回路(インターロック実行部)



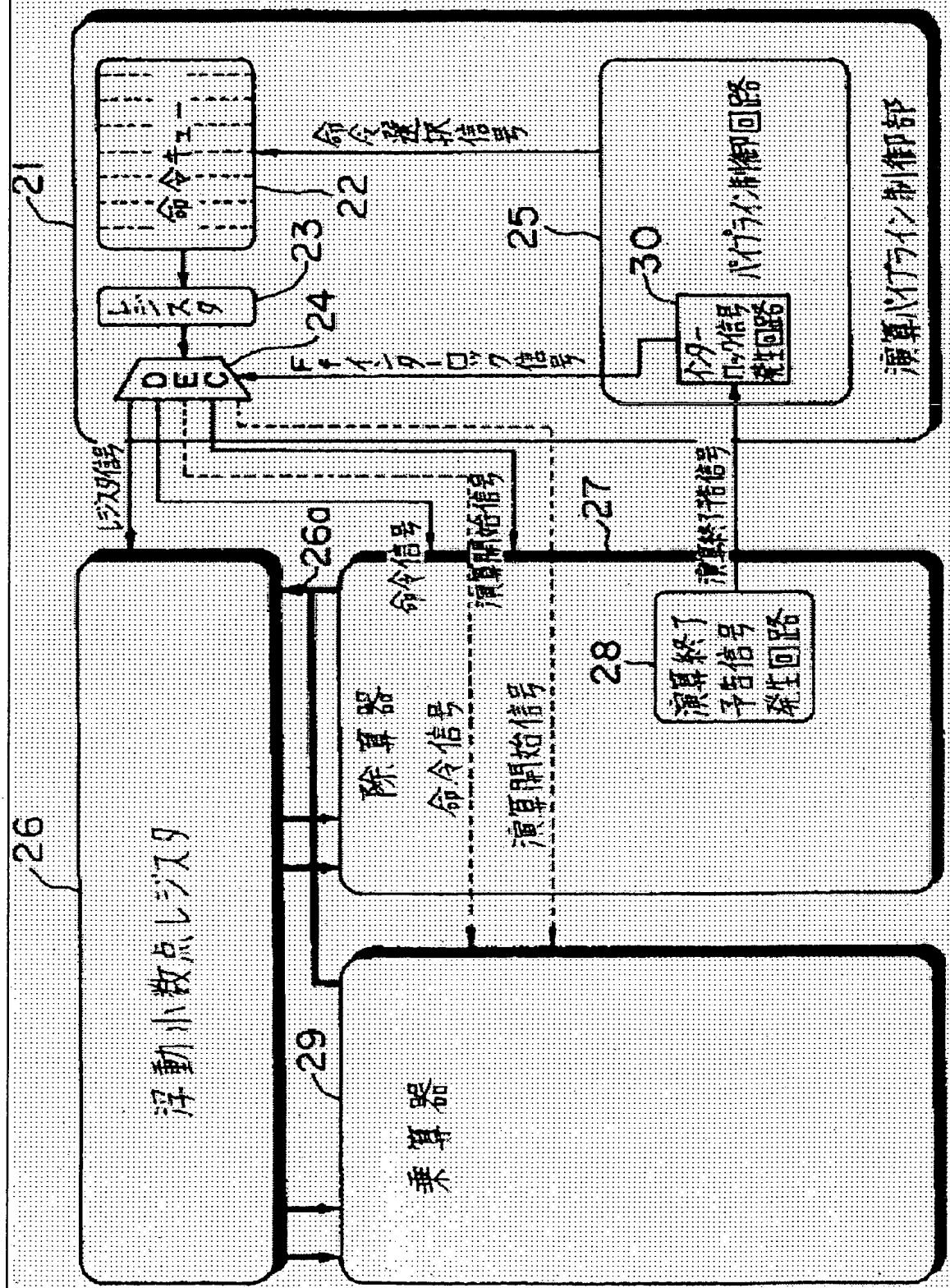
【図1】

## 本発明の原理ブロック図



【図2】

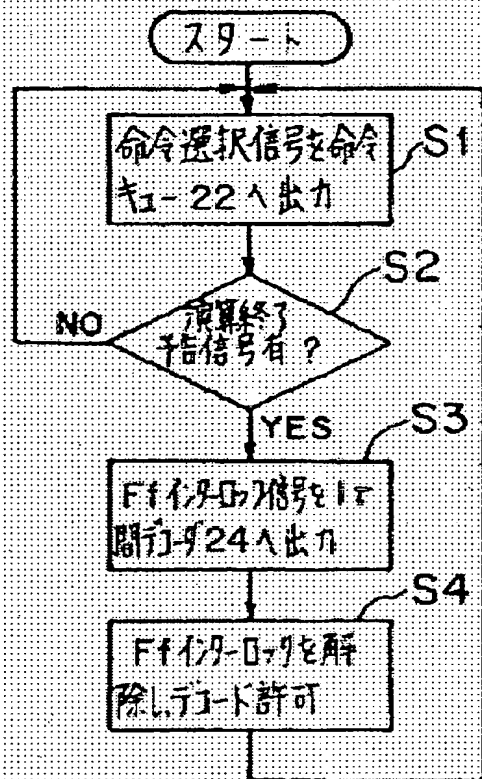
本発明の一実施例としてのパイプライン演算装置を示すブロック図



【図3】

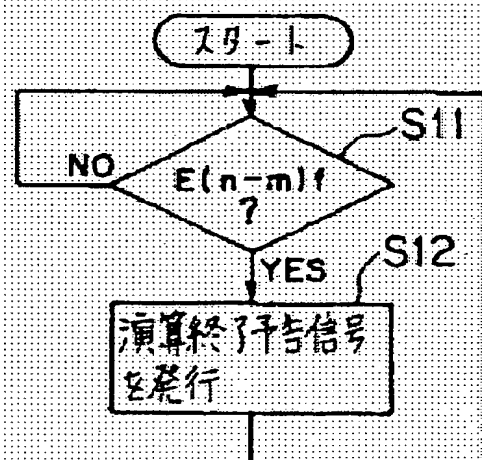
本実施例のパイプライン制御回路の動作を説明するためのフローチャート

(a)

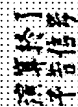


本実施例の演算終了予告信号発生回路の動作を説明するためのフローチャート

(b)



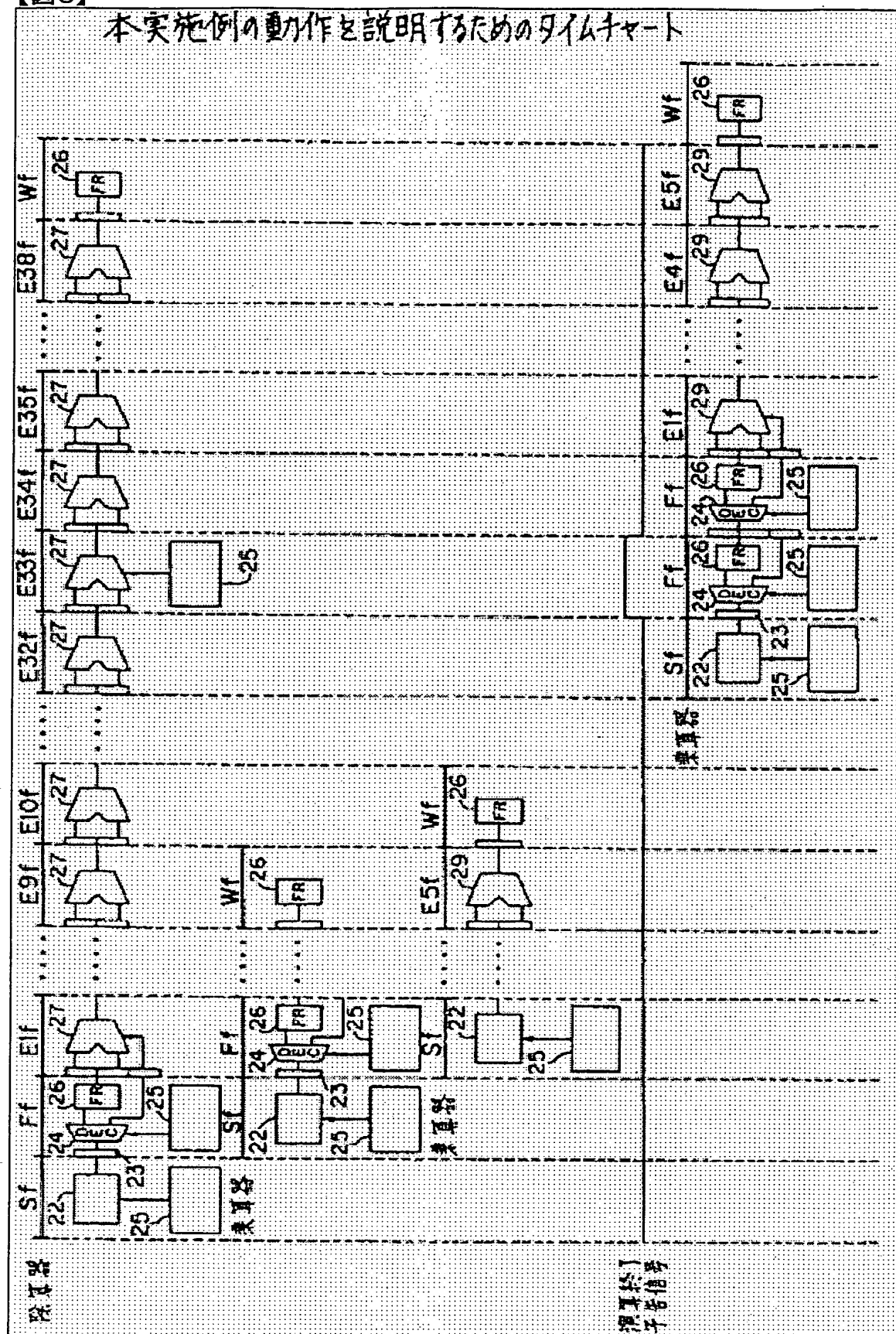
本実施例の動作を説明するためのタイムチャート





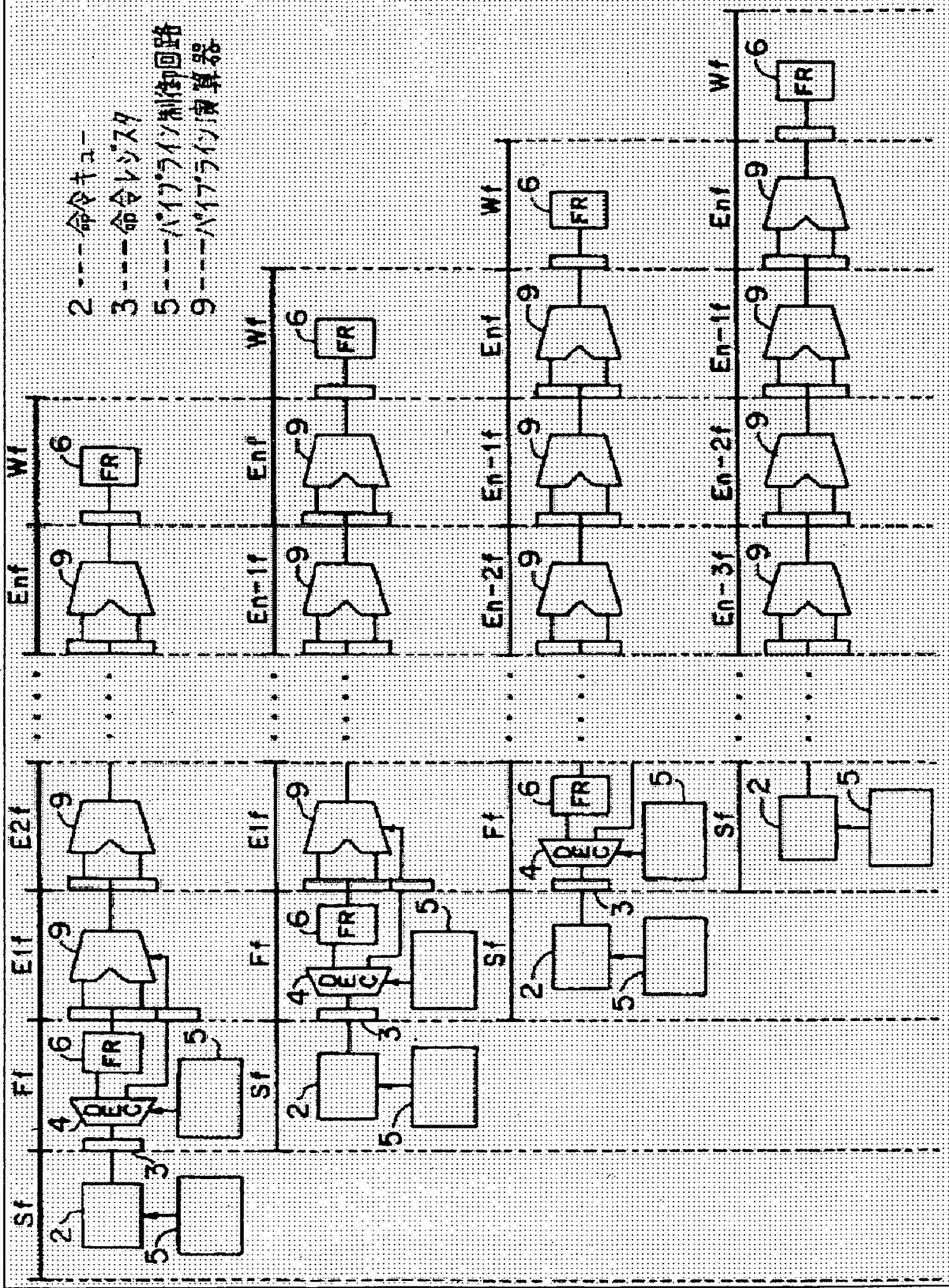
【図5】

本実施例の動作を説明するためのタイムチャート



【図6】

一般的なパイプライン動作を説明するためのタイムチャート

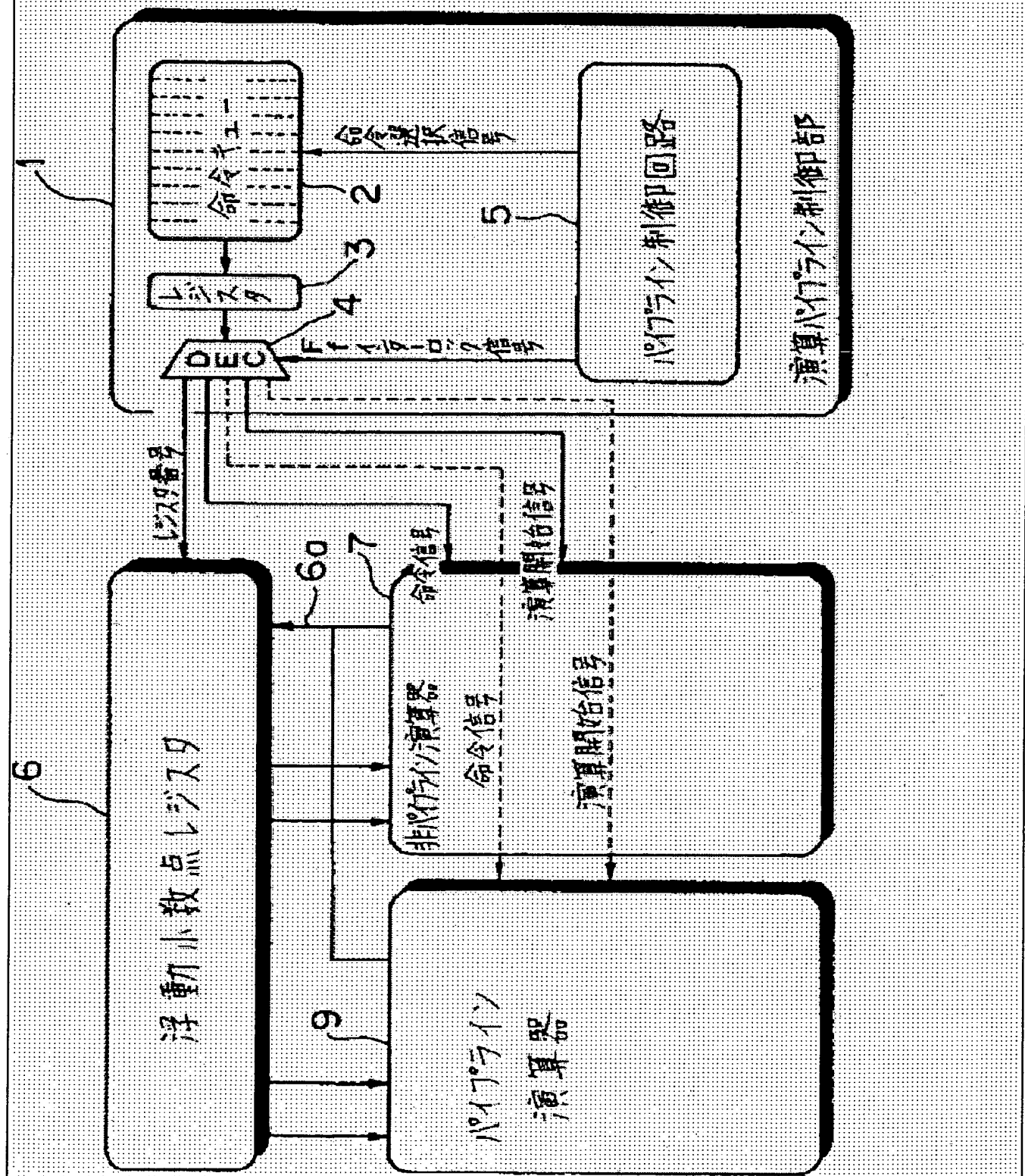


一般的な非パイプライン動作を説明するためのタイムチャート



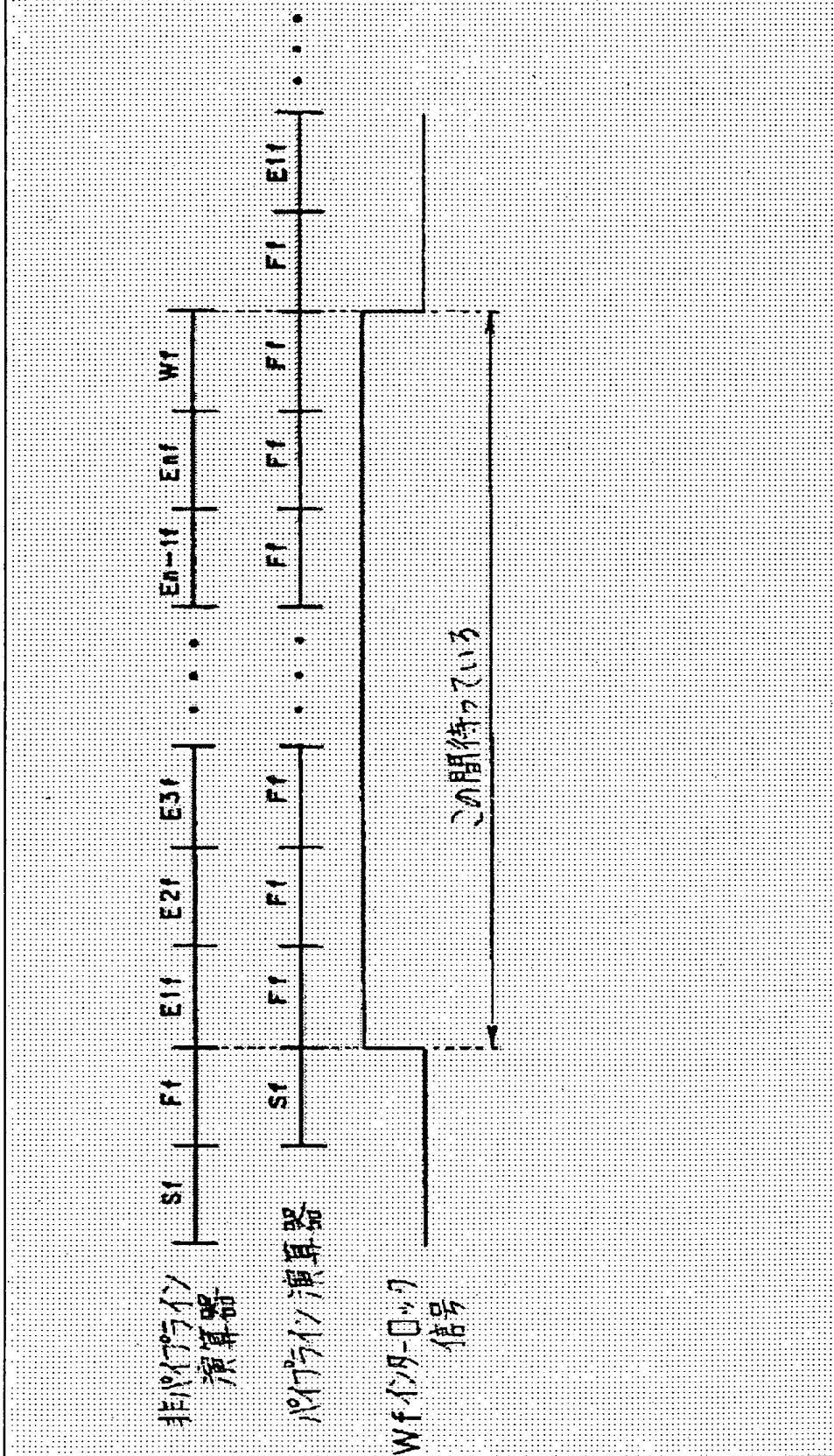
【図8】

パイプライン演算器と非パイプライン演算器とにより1つの書込ポートを共有するパイプライン演算装置の一例を示すブロック図



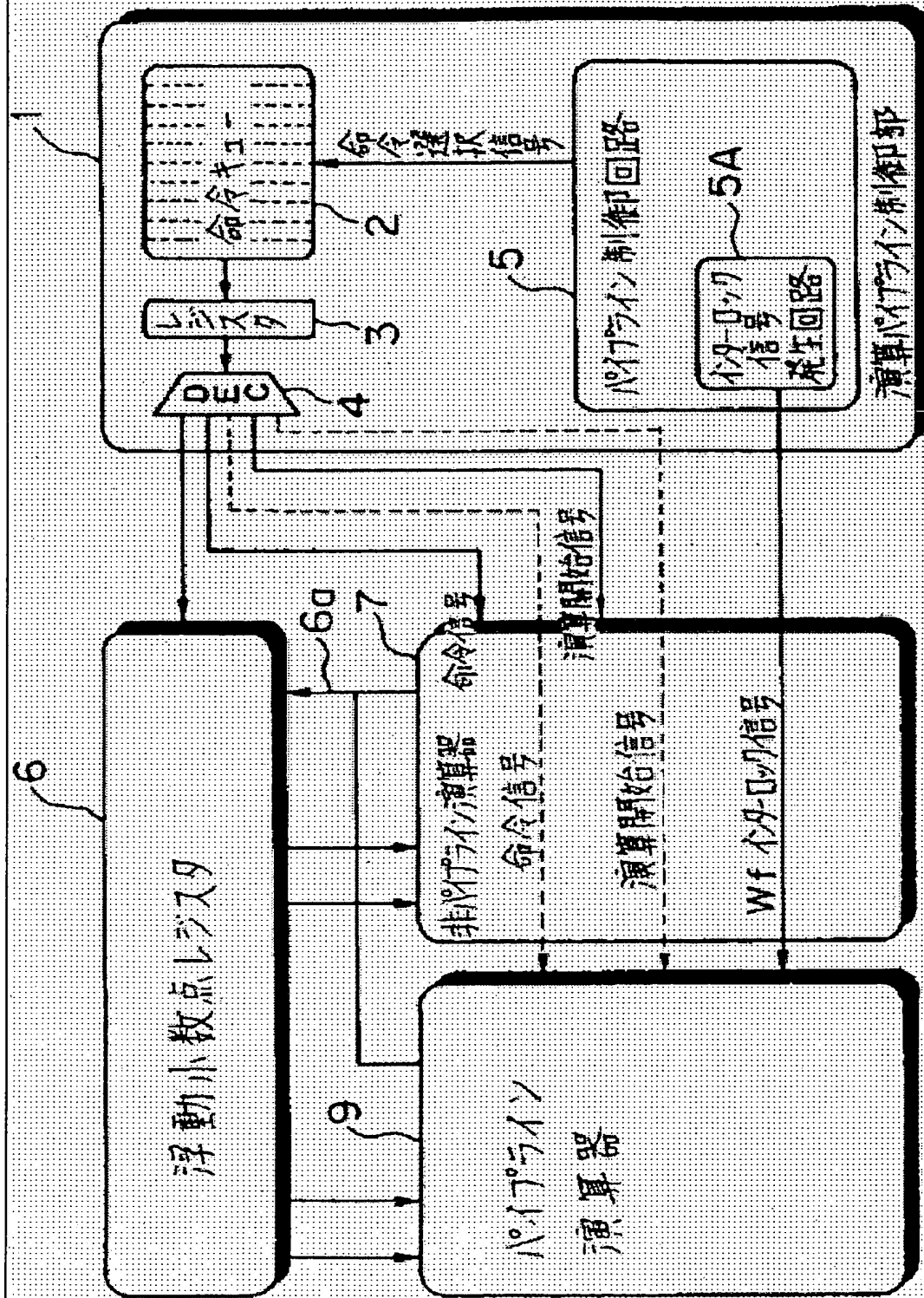
【図9】

図8に示すパイライン演算装置の動作を説明するためのタイムチャート



【図10】

パイプライン演算器と非パイプライン演算器とにより1つの書込ポートを共有するパイプライン演算装置の他例を示すブロック図





【図11】

図10に示すパイプライン演算装置の動作を説明するためのタイムチャート

